CMPT 280

Topic 3: Cursors

Mark G. Eramian

University of Saskatchewan

References

- Textbook, Chapter 3

Cursor Interface

Do you remember what the methods in the cursor interface do?

- `itemExists`
- `item`
- `goFirst`
- `goForth`
- `goLast`
- `goBefore`
- `goAfter`
- `before`
- `after`

Exercise 1

- Write a java interface for the cursor methods (don't worry about the javadoc for now).

```
public interface Cursor<I> {

    /**
     * Checks whether the cursor is positioned at an element in the collection.
     * @return true if the cursor is positioned at an element, false otherwise.
     */
    boolean itemExists();

    ...
}
```

Exercise 2

Write the methods that we need for iterating all of the elements in a linked list. We need:

- `itemExists`
- `item`
- `goFirst`
- `goForth`

Do we need any additional instance variables to support these methods?

Note: While we don't, strictly speaking, need to, the implementation of `goForth` is a lot easier if you've already implemented `after` and `before`.

Additional Pracatice

On your own, implement the remaining cursor methods. You can find the method stubs in the `LinkedList.java` file in the Lecture03 exercise solutions.

- `goLast`
- `goBefore`
- `goAfter`

You could also practice regression testing by writing the regression tests for all of the cursor methods.

Exercise 3

- Using the `LinkedList` class, starting with an empty list, store 5 random numbers in the list, and then iterate over the list, and print out each number.
- Hint: `Math.random()` returns a random `Double` between 0.0 and 1.0.

Exercise 4

- How would we record a cursor position for an `ArrayedList`?
- What additional instance variable(s) must be added to `ArrayedList` to implement the cursor?
- How could we represent the "before" and "after" positions?
- Let's code the required instance variables to `ArrayedList`...
- Do any existing methods need to be changed?

Exercise 5

- Implement the cursor methods for `ArrayedList`
- Additional Practice: On your own, write regression tests for the cursor methods of `ArrayedList`.

Next Class

- How do we know if we've implemented our classes efficiently?
- Next class reading: Chapter 4: Algorithm Analysis.